

Laser Powder Bed Fusion Parameter Selection Via Machine Learning Augmented Process Modeling

Sandeep Srinivasan¹, Brennan Swick², and Michael A. Groeber^{1,*}

¹ The Ohio State University, Department of Integrated Systems Engineering, Columbus, OH 43210

² The Ohio State University, Department of Electrical and Computer Engineering, Columbus, OH 43210

* Corresponding author

E-mail: groeber.9@osu.edu

Abstract

Laser powder bed fusion (LPBF) additive manufacturing (AM) is a highly active research area in the materials and manufacturing community, driven by promises of reduced lead time, increased design flexibility, and potentially location-specific process control. However, a complex processing space counters these benefits and results in difficulties when attempting to develop process parameter sets across different component geometries and sub-geometries. We develop a procedure for coupling physics-based process modeling with machine learning and optimization methods to accelerate searching the AM processing space for suitable printing parameter sets. We demonstrate the approach first on simple geometries that vary in size to show the methodology and then to a more complicated geometry to show the benefit of locally-tailored process parameters on component processing history.

Keywords: additive manufacturing, machine learning, process modeling, optimization

1. Introduction

Laser powder bed fusion (LPBF) additive manufacturing (AM) is currently an intense area of focus within the materials processing community, specifically for metals [1-4]. The advantages of AM include expanded envelopes for design engineers and a more agile supply chain capable of delivering components with shorter lead times. A key feature of LPBF is its locally-resolved processing history, driven by the manner in which the laser energy source is rastered over the powder bed. This *scan path*, defined by upwards of millions of digitally encoded vectors, yields spatially discrete process histories that form a manifold in a complex, high-dimensional processing space with dimensions of space, time, and temperature. The morphology of the local processing manifold can be indirectly manipulated by modifying scan path parameters, such as incident beam energy, beam velocity, beam focus, general raster pattern etc. While this level of processing detail and control offers the potential for tailored location-specific properties, significant difficulties exist in understanding the mapping between both the scan parameters and the high-dimensional processing space, as well as the processing space and the resulting material microstructure.

The mapping of scan parameters to the processing space is further convoluted by the component geometry; features such as corners and edges truncate the scan path, resulting in variation of process history due to the interaction between the incident laser and the component topology. Given the range of geometries being considered for AM, it is critical to be able to quickly identify scan parameters that produce desirable local processing histories. It is highly likely that scan parameters will need to be modified not only between components, but also within a given component. In order to quantify the effect of scan parameter modification on the complex processing space, it is desirable to reduce the overall processing space into a more

compact representation. Previous work has looked at dimensionality reduction of local processing history and *zoning* of components [5]. Zoning is the procedure of labeling regions of a component such that areas with similar processing are categorized under the same label. Ideally, “processing” refers to the most parsimonious space representing the active latent variables that describe the resulting microstructure, which may include aspects of the spatio-temporal thermal fields in an AM build. However, given that such a space is not currently known for AM for many aspects of microstructure, at least not completely, this work will focus only on identifying scan parameters that produce specific local processing zones and not specific microstructures. More specifically, we will focus on the basic, but non-trivial task of producing a set of scan parameters that can be applied locally to a component with complex geometry to yield a minimal number of zones. Effectively, we will attempt to produce a component with a more homogenous processing history.

We present a methodology for 1) using physics-based simulations to produce locally-resolved thermal histories of AM components printed with given scan parameters, 2) reducing the dimensionality of those thermal histories, 3) quantitatively comparing the compact representations to a reference target, and 4) efficiently searching the scan parameter space to find parameter sets that optimally meet conditions of process similarity and printing speed. Specifically, we consider LPBF of a relevant aerospace alloy, Titanium-6wt%Aluminum-4wt%Vanadium (Ti-6Al-4V). We present the methodology first on simple component geometries of varying size to illustrate the effect of scan parameters on processing history. Afterwards, we demonstrate local scan parameter modification within a complex component to benchmark the ability to homogenize processing history within typical AM component geometries. The bulk of this paper is dedicated to the framework developed and not the specific processing conditions being targeted, which are chosen arbitrarily for demonstration purposes.

2. Machine Learning Augmented Modeling Workflow

In this work, we present a framework for identifying scan parameter sets that produce similar local processing histories independent of component geometry. The framework, which augments a physics-based model of the LPBF process, has several key machine learning modules/methods including 1) calibrating the model over the relevant processing space, 2) reducing the model output data, and 3) searching for optimal parameters. The high-level workflow of the framework is presented in Figure 1 and each of the key modules are discussed in more detail in the following subsections.

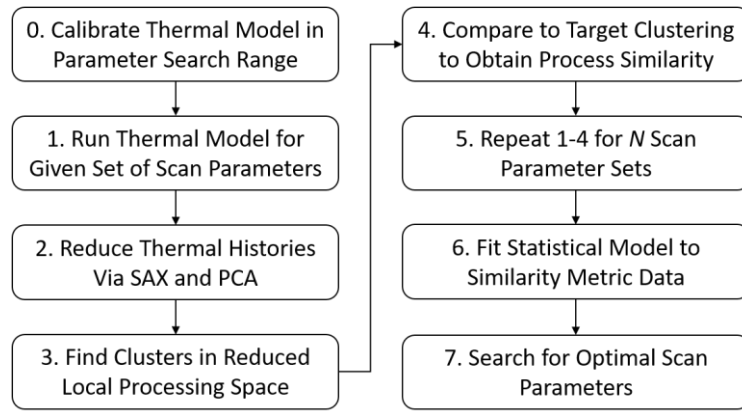


Figure 1: Workflow diagram of the key steps in the scan parameter identification process developed in this work.

2.1 Analytical Thermal Model

Local thermal history is an attractive representation of process state since it simultaneously captures the thermophysical properties of the material along with the incident energy source parameters, including scan velocity and beam energy. Additionally, the local thermal histories are directly influenced by the scan path itself and how that scan path interacts with the component geometry, thereby eliminating the need to explicitly parameterize geometry. To compute the local thermal histories, we utilize an analytical model based on a discrete source representation [6], though other modeling methodologies could be substituted. The

continuously moving energy source is approximated as a series of Gaussian point sources of the following form:

$$s_i(R_{ij}, t) = \frac{A_i \delta(t - \tau_i)}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{R_{ij}^2}{2\sigma^2}\right) \quad (1)$$

where $A_i \delta(t - \tau_i)$ represents the power of the source with the Dirac delta function, δ ; t is time with τ_i representing the impulse time of the energy source; σ sets the spatial size over which the energy is deposited; and $R_{ij} = |\vec{r}_j - \vec{r}_i|$, where \vec{r}_i is the incident source position and \vec{r}_j is the location effected by the source at \vec{r}_i . Integrating (1) over the half space ϑ : $-\infty < x < \infty, -\infty < y < \infty, z \leq 0$ yields the following:

$$\frac{A_i}{(2\pi\sigma^2)^{3/2}} \int_{-\infty}^{\infty} \delta(t - \tau_i) \int_{\vartheta} \exp\left(-\frac{R_{ij}^2}{2\sigma^2}\right) d\vec{r}_j dt = \eta_i P_i \Delta t \quad (2)$$

where P is the power of the incident source and η is an efficiency term. Equation (2) can be expanded to an elliptical Gaussian point source, substituted into the thermal transport equation, and solved using a Green's function approach to yield a relationship for the temperature at location \vec{r}_j at time t due to the source at \vec{r}_i :

$$T(\vec{r}_j, t) = T_0 + \sum_i^N \left[\Theta(t - \tau_i) \frac{\eta_i P_i C}{\sqrt{\lambda_{xi} \lambda_{yi} \lambda_{zi}}} \right] \times \exp\left(-\frac{x_{ij}^2}{2\lambda_{xi}} - \frac{y_{ij}^2}{2\lambda_{yi}} - \frac{z_{ij}^2}{2\lambda_{zi}}\right) \quad (4)$$

where T_0 is the spatially uniform initial temperature; Θ is the Heaviside step function; $C = \Delta t / \sqrt{2\pi} \pi^{3/2} \rho c_p$; $\lambda_{qi} = \sigma_{qi}^2 + 2\alpha(t - \tau_i)$ for $q = x, y, z$; and $q_{ij} = q_j - q_i$ for $q = x, y, z$. A complete derivation of the discrete source approach, along with assumptions and validation

studies may be found in [6]. The calibration methodology for η and σ is discussed in the next subsection.

The discrete source approach has several advantages over other modelling techniques. Representing the incident energy source as a series of superposed events allows for arbitrarily complex scan strategies to be simulated, including point source melting, area melting, or multi-beam approaches. Unlike other analytical methods, such as the classical Rosenthal solution, the discrete source method captures the geometric features of the scan path, such as scan vector length and raster characteristics, in its thermal solution. However, there are several limitations of the discrete source method. For example, it does not directly capture any melt pool dynamics such as waves, sloshing, or convection. Additionally, direct interaction with the powder is not modelled. Choosing to represent processing space as solely thermal history also neglects other features that may influence the resulting microstructure. For example, denudation of powder due to convection in the vapor or occlusion of the laser beam by spatter may lead to lack of fusion defects on subsequent layers, but such occurrences are not captured by thermal history alone [7]. Nevertheless, thermal history provides a concise representation of metal AM processing, and is adequate for the purposes of searching energy input strategies for obtaining process equivalence.

2.2 Regression-based Model Fitting

The two key calibration parameters for the analytical thermal model include an overall energy input efficiency η , and a shape factor expressing the ratio of the depth of the volumetric energy source σ_z to its width σ_x as further described in [6]. The selection of these two parameters is done to best match the cross-sectional meltpool size and shape perpendicular to the moving source, due in part to the importance of meltpool size and shape and the transition to a conduction dominated problem outside of the meltpool. These quantities can be sensitive to the processing parameters, notably in a non-linear fashion near the conduction to keyhole transition

[8]. In order to appropriately account for this dependence, the model must be calibrated against empirical single track results. This calibration must be done for every combination of laser power, laser velocity, laser focus, and powder layer thickness. However, we present a two-step process for obtaining statistical model for each calibration parameter over a range of processing conditions.

First, the thermal model is run for a single track geometry using a wide range of input parameters including power P , velocity v , efficiency η , and depth to width ratio σ_z/σ_x . Layer thickness and beam focus also affect the size and shape of the melt pool, but both have been held constant in this work. The resulting steady state melt pool width w and depth d are determined for each of these cases. Next, response surfaces are fit to create general mappings $f_1 = \eta(P, v, w, d)$ and $f_2 = \sigma_z/\sigma_x(P, v, w, d)$. Note that here w and d are considered input parameters needed to predict the unknown calibration parameters η and σ_z/σ_x . These functions are referred to as the ‘full’ model, since all input parameters are known by running the model. Next, f_1 and f_2 are used in concert with experimental data [6] to predict the η and σ_z/σ_x calibration values required to reproduce the empirical single track observations. A second “direct” response surface is then fit to this dataset which does *not* include the observed width or depth as inputs. Thus, we are left with functions $f_3 = \eta(P, v)$ and $f_4 = \sigma_z/\sigma_x(P, v)$ using the same methodology. This second set of functions can finally be used to predict applicable efficiency and sigma ratio values for arbitrary P, v conditions where analogous single track data is not available. This latter aspect is of critical importance when searching a wide range of processing conditions, while trying to limit extensive experimental work.

There are many suitable methodologies for determining appropriate response surfaces. In the present work, Support Vector Regression (SVR) was employed due to its ability to handle general non-linear functional relationships. The scikit-learn python module was employed for fitting and testing, and radial basis functions were used throughout [9]. All input and output

data were first standardized to have zero mean and unit variance. For the “full” model, 4000 total combinations of P , v , η , and σ_z/σ_x . This data was partitioned into an 80/20 split for the training and testing data respectively. Optimal hyperparameter values for the SVR, including the penalty term C , kernel coefficient γ , and error margin ε were determined using a randomized search cross validation approach. 20 combinations of these values were drawn from exponential distributions and 10-fold cross validation was applied for the full model to select optimal values. A final residual sum of square errors was computed using the reserved test set and was generally observed to be >0.9 for both efficiency and sigma ratio.

For the direct fit, there were only 9 unique P, v combinations, one of which had 3 replicate measurements. Due the limited data, we used no test-train split; instead, all data was used for training. A similar procedure for hyperparameter search was carried out, except 200 combinations were drawn. A residual sum of square errors on the prediction of the full data set was computed to be 0.972 and 0.907 for η and σ_z/σ_x respectively. Figure shows the fit generated for the second model as well as the R^2 value. These values are not a true predictive test of the model since the test and train set are equivalent, but this does confirm that the models reflect the input data.

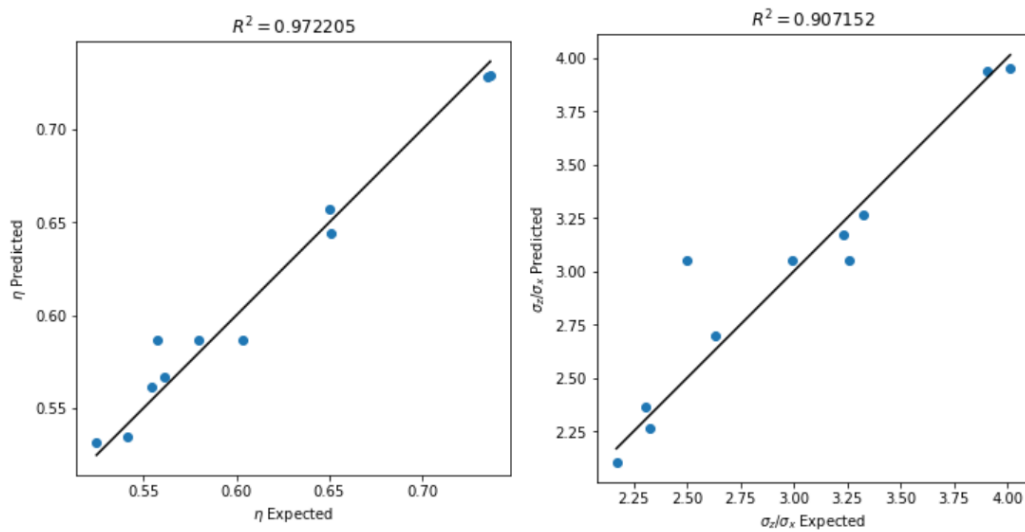


Figure 2: Efficiency (left) and Sigma Ratio (right) fits for the second model

Finally, we note that the single track set used to build the direct model includes data points with $218 \leq P \leq 363\text{W}$, and $908 \leq v \leq 1860\text{mm/s}$; these conditions produced tracks that ranged in width $122 \leq w \leq 177\mu\text{m}$ and depth $86 \leq d \leq 150\mu\text{m}$ as reported in [6]. Based on these empirical observations, the direct model fitting procedure produced predictions with $0.50 \leq \eta \leq 0.75$ and $2.0 \leq \sigma_z/\sigma_x \leq 4.0$. Conditions that fall significantly outside the range of input parameters P and v , observed track dimensions w and d , or calibration parameters η and σ_z/σ_x should be treated with caution, as they are effectively extrapolations.

2.3 Dimensionality Reduction of Thermal Histories

The thermal histories produced from the analytical thermal model may be arbitrarily long, depending on the time step size and window over which the thermal profile is computed. This leads to an exceedingly large dimensionality for a given simulation; over 10 million for a simulation with 10,000 evaluation locations, each with 1,000 time steps in their thermal history. Dimensionality reduction can be used to produce a more tractable representation of the thermal histories for use in further analysis. Dimensionality reduction techniques embed the high-dimensional processing space into a lower dimension, while attempting to avoid overly distorting the data [10]. Several classical and advanced techniques exist for performing dimensionality reduction, including principal component analysis (PCA) [11, 12], isomap [13], autoencoders [14], and laplacian eigenmaps [15–16], and may fit linear or nonlinear embeddings.

Since thermal histories are a kind of time series data set, we have chosen a technique that is explicitly formulated for time series, an approach called symbolic aggregate approximation (SAX) [17]. SAX maps a time series, represented as a piecewise aggregate, to an alphabet of predetermined size to produce a string representation of the data [17]. Standard approaches for representing time series in reduced form usually rely on decomposing the data into a linear sum of basis functions, determined by methods like a Fourier transform. The basis functions of SAX

can be thought of as box waves, where the amplitude of each box wave determines its alphabet mapping. To produce a symbolized representation of this piecewise aggregate, a set of breakpoints on the values of the time series are chosen, which are each mapped to a unique symbol in the chosen alphabet. There are statistical advantages to choosing breakpoints such that any character from the chosen alphabet is equiprobable [17, 18]. We instead choose breakpoints to correspond to equilibrium phase transition temperatures. Thus, when the same character appears in sequence, the material is being held in a specific phase field, and a change in characters represents passing through equilibrium phase transitions. This approach makes the string representations more ingestible for material scientists.

Since we directly map the discretized thermal history to an alphabet, the resulting dimensionality of the symbolic representation is the same as the original profile. Also, thermal histories may be of varying lengths, making distance comparisons between the string representations difficult. We address both these issues by using a technique from Kumar et al. to convert each string representation into a quantized vector of known length depending on the size of the underlying alphabet [19]. This vector is constructed by counting the frequencies of subwords of chosen length within the overall SAX representation. For example, consider an alphabet consisting of **a** and **b**. Using a subword length of two, we count all occurrences of **aa**, **bb**, **ab**, and **ba**, producing a vector of length four whose components are the count frequencies of each subword. Since the breakpoints for the alphabet are chosen to correspond to equilibrium phase transition temperatures, the resulting frequency-based representation yields insights into the type and number of phase transitions traversed by a given thermal history.

For each generated thermal history, we compute the SAX representation using an alphabet with five characters. Three temperature breakpoints are set at 995 °C, 1650 °C, and 3287 °C, to represent the $\alpha \rightarrow \beta$ transformation, melting, and boiling equilibrium phase transition temperatures, respectively, for Ti-6Al-4V. The fourth breakpoint is set at 600 °C, chosen to

approximate a temperature below which kinetic factors are small. We then map each SAX sequence to a fixed-length feature vector by counting pairwise subwords. This yields a 25 dimensional feature for each thermal history, regardless of length, where each dimension is the frequency of the corresponding subword in the SAX representation. Subwords that are composed of different characters represent either heating or cooling, depending on their ordering, whereas subwords that are the same letter represent periods the thermal history remained within a particular temperature window. Figure 3a schematically shows the 25 dimensional subword space. Subwords that differ by more than one letter correspond to temperature jumps of more than one phase transition in the underlying time discretization of the thermal history. We synthetically increase the temporal resolution of thermal history by recursively counting smaller transition subwords. For example, if an **ae** subword is encountered in the SAX sequence, we increment the counters for **ab**, **bc**, **cd**, and **de** instead. The motivation for this approach is to alleviate issues caused by the time resolution of the simulation being too coarse to capture transitions that occur during the rapid heating/cooling events characteristic of powder bed fusion. The underlying assumption is that the thermal history is a continuous function that, by the intermediate value theorem, must pass through all subword sequences when considering temperature jumps. Another motivating factor is to further reduce the overall dimensionality of the subword space. Using this approach results in the total number of non-empty dimensions being 13. Figure 3b shows which subword features appear in the resulting feature space. Finally, we choose to remove the **aa** subword because that temperature domain was chosen to capture time steps when no significant activity is occurring, which reduces the final feature vector to 12 components.

Even 12 dimensions is a relatively large space and for subsequent analyses, we wish to further reduce dimensionality. Principal Component Analysis (PCA) is the technique adopted to reduce the dimensionality of the subword vectors. The feature projection of the subword

vectors i.e., the transformation of the data in higher dimensional space (12-components) to a lower dimensional space (2-components or 3-components), for easy interpretation and visualization was achieved using the orthogonal linear transformation of the PCA. We first standardize the dataset, and then a covariance matrix is created, which is fitted to the PCA module of scikit-learn. The PCA module uses Singular Value Decomposition (SVD) (factorization of a real or complex matrix into singular vectors and singular values), as the metric to project the data onto a lower dimensional space and to calculate the eigenvectors also known as *principal components* along with their corresponding eigenvalues. The eigenvalues and eigenvectors are then sorted in descending order, which ranks the principal components from highest to lowest explained variance. The number of principal components selected determines the amount of variance explained and the dimensionality of the data set. For this work, the number of principal components was deduced using Figure 3c. Approximately 85% of the variance is explained with the first two principal components. The first two components were selected to represent each spatial evaluation location's thermal history because each additional principal component accounted for less than 10% of the explained variance.

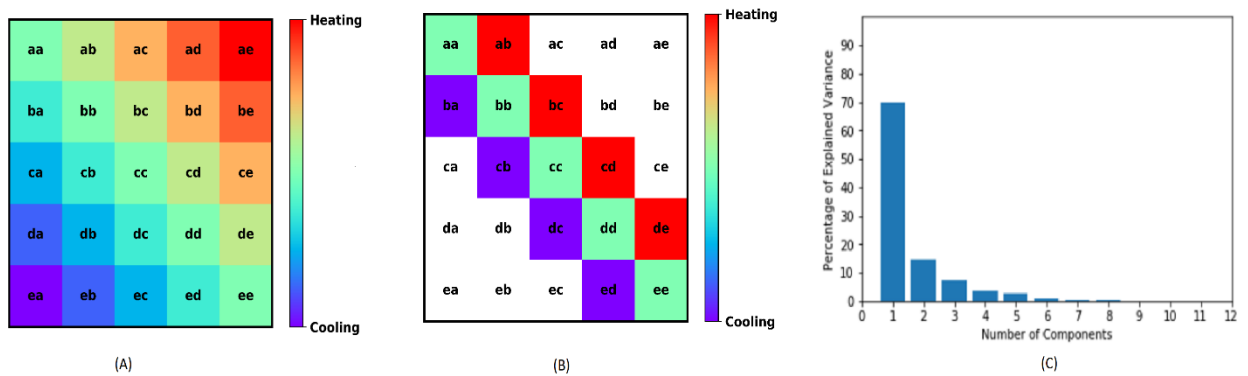


Figure 3: (A) Full 25-component subword space, (B) Reduced 13-component subword space, and (C) plot of the percentage of variance explained by each principal component determined by SVD of the reference data set.

2.4 Clustering of Reduced Thermal Histories

The previous discussion of dimensionality reduction was focused solely on individual thermal histories. However, even along a single vector in an AM build, there is variation in thermal history due to the macro scan strategy and geometry. As such, even if each spatial location's thermal history is now represented by two values, there will be a spread associated with those two values. This is evident in Figure 4a, which shows 10,000 spatial locations, represented by two values, within a 15 mm square patch printed with our selected "reference" scan parameters (300 W laser power, 1300 mm/s laser velocity, and 0.14 mm hatch spacing). To further reduce the dimensionality (20,000) associated with the spatial heterogeneity of AM process history, we utilize methods from cluster analysis, which can be used to partition a data space into groups. Specifically, we employ the widely popular Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [20]. The DBSCAN algorithm, unlike K-means (another common clustering method), is a non-parametric clustering algorithm since it does not make any assumptions on the population distribution and the sample size required to develop a model. Given a set of points in some space, DBSCAN works to group together points that are closely packed together and marks outlier points that lie alone in low-density regions. Controlling parameters include epsilon, which sets the neighborhood radius, and the minimum number of points needed to establish a cluster.

After DBSCAN is applied to a given set of local thermal histories, we obtain a labeling with all points mapped to one of N clusters. In order to reduce the dimensionality of the distribution, we develop a five-component feature vector for each cluster identified in the reduced local processing space, containing: the PC1 and PC2 coordinates of the centroid of the cluster, the PC1 and PC2 variances of the states within the cluster, and the fraction of all states that belong to the cluster. Thus, for the clustering of a given data set, there is a $5N$ -component feature vector that defines the distribution of local processing states. As a result, the original 10+

million dimensional representation of local processing states has been reduced to ~ 25 , depending on N . Figure 4a-d shows the DBSCAN results for a reference data set and three other data sets compared to the reference.

2.5 Similarity Metric Definition and Fitting in Scan Parameter Space

Once a $5N$ -component feature vector is obtained for the clustering of some reference domain, we must develop a metric for comparing the similarity of any other clustering in the reduced local processing space produced by a different scan parameter set or geometry or both. We can compute an analogous $5N$ -component feature vector for any other distribution of local processing states produced by a different scan parameter set. We then define a relatively simple similarity metric as the summed 5d Euclidean distance of each cluster in the current distribution to its closest analogous cluster in the reference distribution. The metric is given by the following:

$$\text{similarity metric (SM)} = \sum_i^{N^C} \sqrt{\sum_j^5 (X_{ij}^C - X_j^{R*})^2} + 1000 * \begin{cases} N^R - N^C & \text{if } N^R > N^C \\ 0 & \text{else} \end{cases} \quad (5)$$

where N^C is the number of clusters in the current distribution, N^R is the number of clusters in the reference distribution, X_{ij}^C is the j^{th} component of the 5d feature vector of the i^{th} cluster in the current distribution, and X_j^{R*} is the j^{th} component of the 5d feature vector of the closest cluster in the reference distribution to the i^{th} cluster in the current distribution. The clustering of the two distributions is done independently, but using the same DBSCAN controlling parameters, which can result in different numbers of clusters in the two distributions. To account for this, if the current distribution has more clusters, then multiple clusters will pair with the same cluster in the reference and the metric will inherently carry the penalty of

summing more distances. However, if the current distribution has fewer clusters, we add a penalty for each missing cluster (seen in the second term of equation 5).

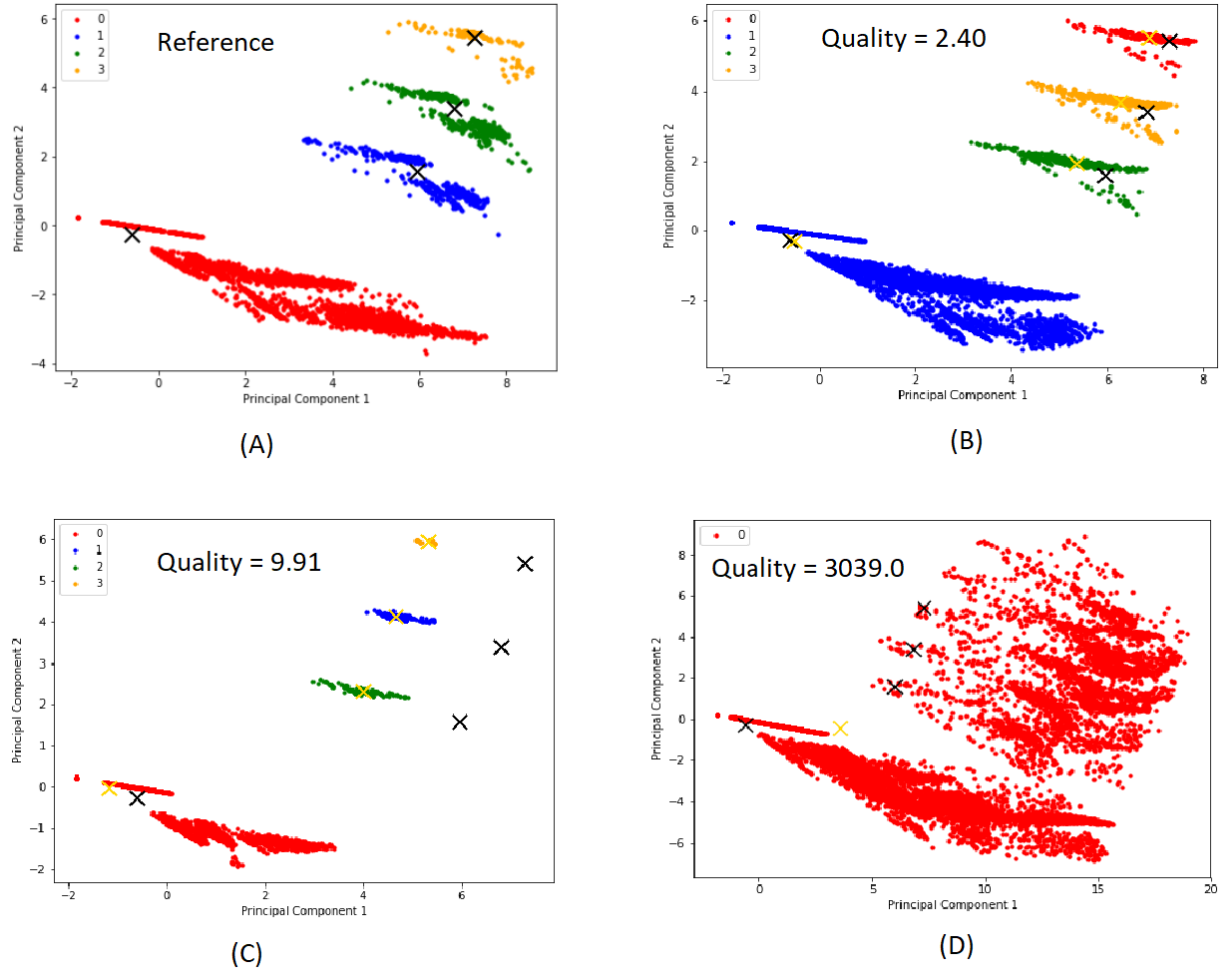


Figure 4: (A) DBSCAN clustering of 15mm reference tile printed with scan parameters (300W, 1300mm/s, 0.14mm hatch spacing), and clustering results and similarity metric of 9mm tile printed with (B) 375W, 1400mm/s, 0.2mm hatch spacing, (C) 350W, 1500mm/s, 0.25mm hatch spacing, and (D) 350W, 1100mm/s, 0.14mm hatch spacing. In each of (B-D) the black “x”s represent the centroids of the clusters of the reference tile in (A).

Ultimately the goal of this work is to minimize the similarity metric by changing scan parameter sets locally within a component. Thus, we need to be able to obtain the similarity metric at all points within the scan parameter space of interest. In order to obtain the similarity metric, it is necessary to run the calibrated thermal model for the current scan parameter set, reduce the dimensionality of the resultant thermal histories, cluster the distribution of local processing states, and then compute the metric in equation 5. This is relatively expensive

computationally, especially if fine steps in scan parameter space is desired. For this work, we attempt to improve the speed associated with obtaining a similarity metric for a given scan parameter set by proposing a regression approach, similar to the one used in the model calibration process. We perform a grid sampling of scan parameter space and compute the similarity metric via the loop described above. We then perform SVR, in the same manner as the fitting of the model calibration parameters in section 2.2, to obtain a general mapping $f = SM(P, v, hs)$ that can be used as a fast-acting statistical model for obtaining the similarity metric given a scan parameter set. For this work we chose a somewhat coarse sampling of the processing space with steps of 25 W in laser power, 100 mm/s in laser velocity, and 0.01 mm in vector or hatch spacing. The resultant SVR fits were acceptable, but not as high quality as the model fitting in section 2.2, which may be attributed to the discrete penalty we add for missing clusters causing sharp gradients in the similarity metric. The SVR fits generally had an R^2 value of ~ 0.8 .

2.6 Process Parameter Search Using A Multi-Objective Gradient-Based Approach

When searching for a local scan parameter set, we define a multi-objective goal that aims to ensure a threshold similarity metric, while minimizing the time to print ($t = L^2 / (v \cdot hs)$, for a square tile). In order to search scan parameter space for an optimal parameter set, we implement a gradient descent approach using the SVR-obtained mapping $f = SM(P, v, hs)$. P and v were constrained by $200 \leq P \leq 375\text{W}$, and $1000 \leq v \leq 2000\text{mm/s}$, respectively, because the direct model cannot reliably predict data outside of these ranges. The hatch spacing, hs , was constrained by $0.10 \leq hs \leq 0.25\text{ mm}$ as these values closely match the extremes of single track width predicted by the model in the aforementioned laser power and speed ranges. A threshold for the similarity metric was obtained by examination of the values obtained by the initial grid search, shown in Figure 5a. It can be seen in the plot, which rank-orders the similarity metric values, that there are large steps in the similarity metric associated with

discrete clusters being identified by the DBSCAN algorithm. In general, only when the correct number of clusters exist in the current distribution does the similarity metric reach a low value, which is seen in the zoomed-in inset in Figure 5a. For this work, we arbitrarily chose a similarity metric threshold of 5 to define acceptable similarity, which is noted in Figure 5a.

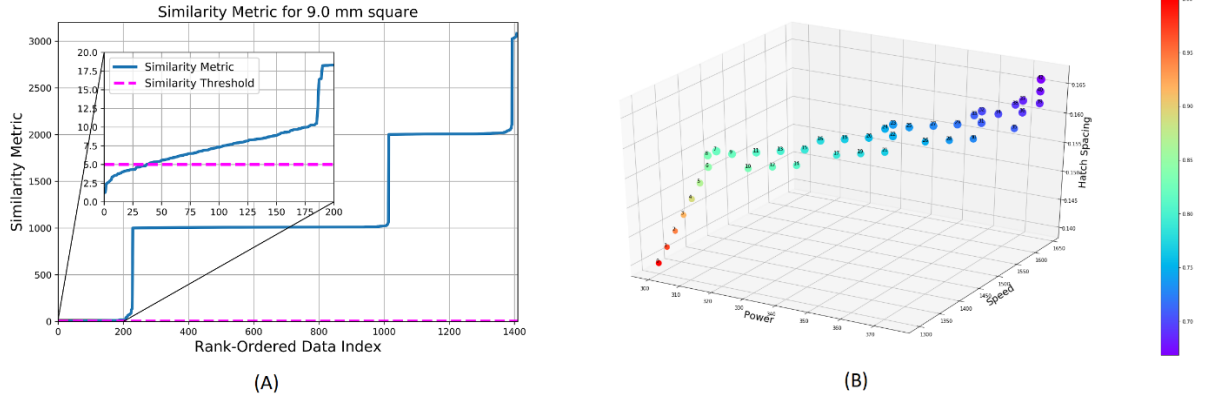


Figure 5: (A) plot of a rank-ordered list of similarity metrics calculated during a grid search of scan parameters applied to a 9mm tile. Similarity threshold shows acceptable similarity metric used during optimal parameter search. (B) trajectory plot showing the evolution of scan parameters during the gradient descent-driven optimization process for the 9mm tile. The size of the points is linked to the similarity metric and the color represents the fraction of time required to print the tile relative to printing it with the reference tile’s scan parameters.

The gradient descent algorithm was initialized using the P , v , and hs combination from the reference tile as the starting state. Both the similarity metric and time to print were stored at this state before a list of available actions, increase a step-size, decrease a step-size, or remain constant for P , v , and hs was compiled. The step sizes were chosen as 5 W, 20 mm/s, and 0.002 mm respectively for each state parameter, which is a factor of 5 finer in resolution than the original grid search. If any of the available actions caused the state to move outside the constraints, it was removed from the set of available actions. For each available action the similarity metric and time to print were compared to the similarity metric and time to print of

the current state. If the similarity metric of the current state and any of the available states were below the threshold, the action was chosen that minimized the time to print. In all other cases, the action was chosen that minimized the similarity metric. The state was updated, and the process was repeated until the current state was the same as the previous state. Figure 5b shows a typical trajectory of the scan parameter states during the optimization process.

3. Results

In order to demonstrate the utility of our proposed approach, we demonstrate local scan parameter modulation on a single 2D layer of an arbitrary geometry that demonstrates some of the complexity of typical AM components. For simplicity, we designed a component composed of rectilinear subfeatures to make tiling a simple procedure. Components with more complex, curved geometries, would require a more complex scan parameter modulation. The geometry we designed can be seen in Figure 6. The complex geometry was first scanned with the reference scan parameters (300W, 1300mm/s, and 0.14mm hatch spacing), using a *stripe width* of 15mm, which means all vectors are 15mm in length unless truncated by the geometry. Second, the complex geometry was tiled using square tiles of one of five sizes (3mm, 6mm, 9mm, 12mm, and 15mm). The geometry was created in a way that tiles of these five sizes can be arranged such that each are printed in their entirety, which removes the issues associated more complex scan parameter search. The more general tiling and scan parameter search is the topic of future work.

Tile Size (mm)	Laser Power (W)	Laser Velocity (mm/s)	Hatch Spacing (mm)
15 (reference)	300	1300	0.14
12	370	1580	0.176
9	375	1640	0.166
6	325	1440	0.14
3	330	1500	0.174

Table 1: Scan parameter sets identified by the gradient descent-based search algorithm for various sizes of square tiles. The optimized scan parameter sets print the tiles as fast as possible, while ensuring a similarity threshold.

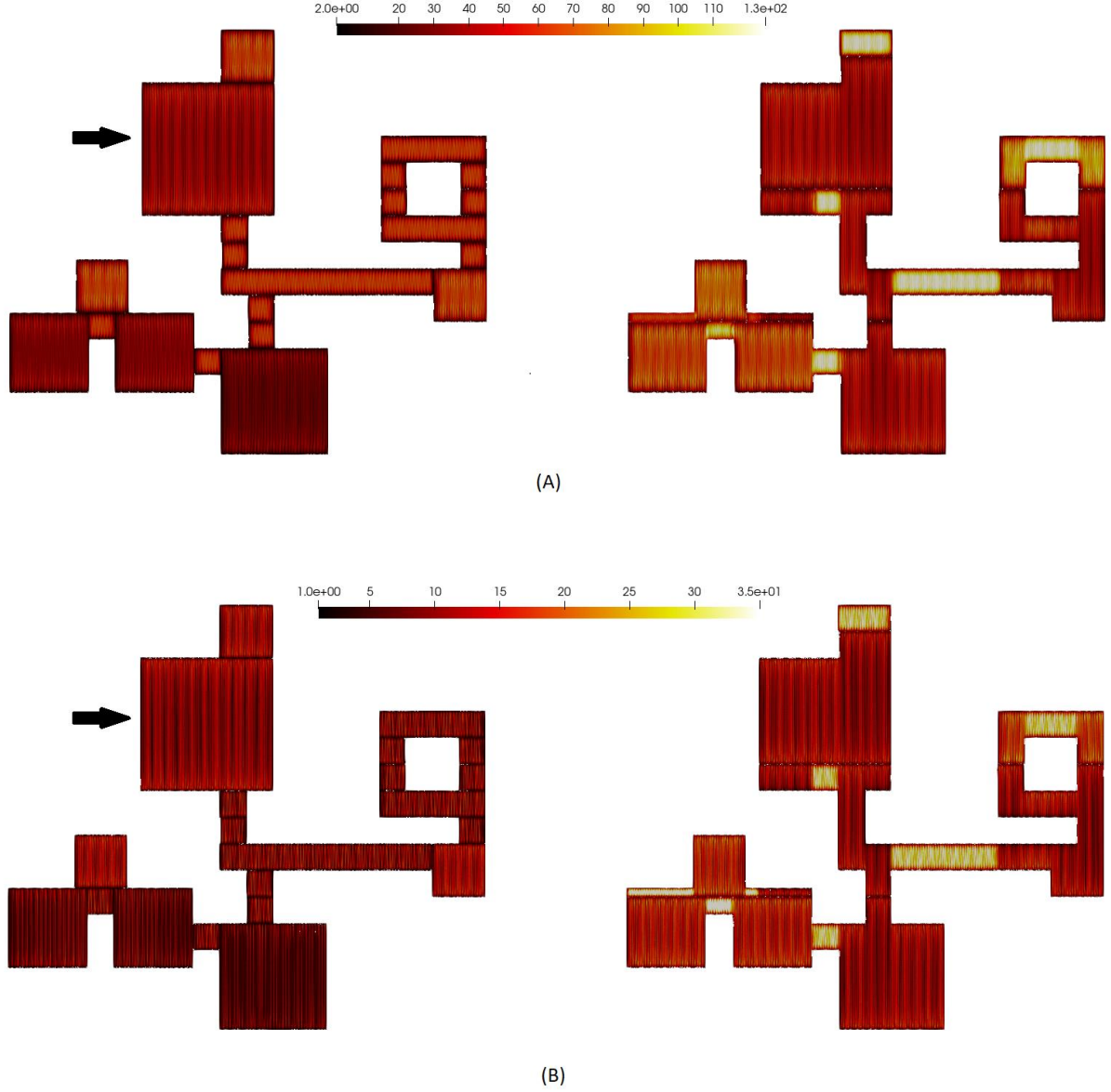


Figure 6: Results showing (A) the time spent in the elevated temperature α phase field (corresponding to the **b** alphabet label) for the uniform (right) scan parameter strategy and the locally-modulated (left) scan parameter strategy; and (B) the number of time steps each location spends in the β phase field (corresponding to the **c** alphabet label) for the uniform (right) scan parameter strategy and the locally-modulated (left) scan parameter strategy.

First, the 15mm tile was printed with the reference scan parameters to obtain the target distribution of local processing states. Next, the scan parameters of each of the other four tile sizes were optimized by the process outlined in the previous sections. The results of that optimization process are shown in Table 1. After obtaining the optimized scan parameters of the tiles, the tiles are positioned to create the larger component geometry and are printed in sequence with their local scan parameters, effectively modulating the scan parameters across the component. The results of the uniform and locally modulated scan parameter strategies are seen in Figure 6a-b. In the figure, we show that both the number of time steps each location spends in the elevated temperature α phase field (corresponding to the **b** alphabet label) and the number of time steps each location spends in the β phase field (corresponding to the **c** alphabet label). It is obvious in the figure that the uniform scan parameter strategy creates significant heterogeneity in the component with some locations differing by almost a factor of 10. In the case of the locally modulated scan parameter strategy, the heterogeneity is significantly reduced with location generally all being within a factor of 2-3. It should be noted that the results in the locally modulated case are not necessarily the best that could be obtained from a heterogeneity minimization standpoint, because the objective of the optimization was only to obtain a threshold similarity while minimizing the printing time, not obtain the most similar processing possible. An additional positive aspect of the optimization is that the full component was actually able to be printed in less time with more uniform processing using this approach.

4. Conclusion

We demonstrate a process for using physics-based models, coupled with machine learning regression methods, dimensionality reduction, and optimization to determine local scan parameters to yield AM components with more uniform local processing. The current work remains relatively simple in the geometries investigated and methods for comparing processing

states, but more complicated geometries and metrics are currently being investigated. The approach shows promising results in the ability to produce more uniform components in reduced printing time.

References

- [1] T.J. Horn, O.L. Harrysson, Overview of current additive manufacturing technologies and selected applications, *Science Progress* 95 (3), 2012, 255-282.
- [2] W.E. Frazier, Metal Additive Manufacturing: A Review, *Journal of Materials Engineering and Performance* 23 (6), 2014, 1917-1928.
- [3] W.J. Sames, F.A. List, S. Pannala, R.R. Dehoff, S.S. Babu, The metallurgy and processing science of metal additive manufacturing, *International Materials Reviews* 61 (5), 2016, 315-360.
- [4] D. Herzog, V. Seyda, E. Wycisk, C. Emmelmann, Additive manufacturing of metals, *Acta Materialia* 117, 2016, 371-392.
- [5] S. Donegan, E. Schwalbach, and M. Groeber, Zoning additive manufacturing process histories using unsupervised machine learning, *Materials Characterization*, 161, 110-123.
- [6] E. Schwalbach S. Donegan, M. Chapman, K. Chaput and M. Groeber, An Analytical approach to Predict Thermal History Produced by Complex Moving Heat Sources, *Additive Manufacturing*, 25, 485-498.
- [7] S.A. Khairallah, A.T. Anderson, A. Rubenchik, W.E. King, Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones, *Acta Materialia* 108, 36-45.
- [8] J. Trapp, A.M. Rubenchik, G. Guss, M.J. Matthews, In situ absorptivity measurements of metallic powders during laser powder-bed fusion additive manufacturing, *Appl. Mater. Today* 9 (Suppl. C), 2017, 341-349.
- [9] F. Pedregosa et al. Scikit-learn: Machine Learning in Python. *J. Machine Learning Res.* 12, 2011, 2825-2830.
- [10] L. van der Maaten, E. Postma, J. van den Herik, Dimensionality Reduction: A Comparative Review, 2009.
- [11] K. Pearson, On lines and planes of closest fit to systems of points in space, *Philosophical Magazine* 2, 1901, 559-572.
- [12] H. Hotelling, Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology* 24, 1933, 417-441.
- [13] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (5500), 2000, 2319-2323.
- [14] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science* 313 (5786), 2006, 504-507.
- [15] M. Belkin, P. Niyogi, Laplacian Eigenmaps and spectral techniques for embedding and clustering, *Advances in Neural Information Processing Systems*, Cambridge, MA, The MIT Press, 14, 2002, 585-591.
- [16] M. Belkin, P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, *Neural Computation* 15 (6), 2003, 1373-1396.
- [17] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, *DMKD '03 Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003, 2-11.
- [18] A. Apostolico, M.E. Bock, S. Lonardi, Monotony of Surprise and Large-Scale Quest for Unusual Words, *Proceedings of the 6th International Conference on Research in Computational Molecular Biology*, 2002, 22-31.
- [19] N. Kumar, V.N. Lolla, E. Keogh, S. Lonardi, C.A. Ratanamahatana, L. Wei, Time series Bitmaps: A Practical Visualization Tool for Working with Large Time Series Databases, *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005.

-
- [20] Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M. (eds.). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231.